台科大 111 學年度「彩色影像處理」作業三：簡化的 CNN 數值影像辨識

M11125016 蕭強

Programming language: Python

```python
import random
import cv2
import numpy as np
import matplotlib.pyplot as plt
from sklearn.metrics import confusion_matrix

data_path = "C:/Users/cghsi/Desktop/HW3/train1000/"

# 第一層捲積
fil1 = np.array([-1, -1, 1, -1, 0, 1, -1, 1, 1]).reshape(3, 3)
# 第二層捲積
fil2 = fil1.T


def readImg(num1, num2):
    """
    讀取照片
    """
    train = []

    num1 = num1 + 1 if num1 == 0 else num1 * 100
    num2 = num2 + 1 if num2 == 0 else num2 * 100

    for i in range(num1, num1 + 100):
        img = cv2.imread(data_path + str(i) + ".png", 0)
        img = img / 255
        img = cv2.resize(img, (8, 8), interpolation=cv2.INTER_AREA)
        train.append(img)

    for i in range(num2, num2 + 100):
        img = cv2.imread(data_path + str(i) + ".png", 0)
        img = img / 255
        img = cv2.resize(img, (8, 8), interpolation=cv2.INTER_AREA)
        train.append(img)
    return train


def con2d_1l(input, kernel1=fil1, kernel2=fil2):
    """
```

```
    第一層卷積
    """
    output = np.array([cv2.filter2D(input, -1, kernel1), cv2.filter2D(input, -1,
kernel2)])
    output = output.transpose(1, 2, 0)
    return output


def con2d_2l(input, kernel1=fil1, kernel2=fil2):
    """
    第二層卷積
    """
    a = np.array([cv2.filter2D(input[:, :, 0], -1, kernel1)])
    b = np.array([cv2.filter2D(input[:, :, 1], -1, kernel2)])
    output = np.concatenate((a, b))
    output = output.transpose(1, 2, 0)
    return output


def maxPooling(feature_map, size=2, stripe=2):
    """
    MaxPooling
    """
    in_ax0, in_ax1, in_ax2 = np.shape(feature_map)
    # output size
    output_ax0 = int((in_ax0 - size) // stripe + 1)
    output_ax1 = int((in_ax1 - size) // stripe + 1)
    output_ax2 = in_ax2
    output = np.zeros((output_ax0, output_ax1, output_ax2))

    for i in range(output_ax0):
        for j in range(output_ax1):
            for f in range(output_ax2):
                start_y = i * stripe
                start_x = j * stripe
                crop_map = feature_map[start_y:start_y + size, start_x:start_x + size, f]
                max_value = np.max(crop_map)
                output[i, j, f] = max_value

    return output


# def confusion_matrix(Y_true, Y_pred):
```

```
#     mat = np.zeros((2, 2))
#     K = len(np.unique(Y_pred))
#     print(K)


def trainModel(dataset, num1, num2):
    # label X is the feature map after convolution
    X = []

    train_nums = len(dataset)
    Y = np.zeros(train_nums)
    Y[int(train_nums / 2):] += 1
    Y = np.array(Y).reshape(train_nums, 1)

    for img in dataset:
        # first layer
        img = con2d_1l(img)
        img = maxPooling(img)
        # second layer
        img = con2d_2l(img)
        img = maxPooling(img)
        # flatten layer
        img = np.ndarray.flatten(img)
        img = np.append(img, 1)
        X.append(img)

    X = np.array(X)
    # Linear regression
    A = np.dot(np.linalg.inv(np.dot(X.T, X)), (np.dot(X.T, Y)))
    # Predict value
    Yp = np.dot(X, A)
    # Confusion matrix
    Y_pred = np.where(Yp > 0.5, num2, num1)
    Y_true = Y
    confusion_mat = confusion_matrix(Y_true, Y_pred)
    acc = confusion_mat[0][0] + confusion_mat[1][1] / train_nums

    print(f"Confusion matrix:\n{confusion_mat}")
    print(f"Accuracy:{acc}")
    return Y_pred, Y_true


def plot_result(label, predic, num1, num2):
```

```
    plt.figure()
    for i in range(15):
        random_img = random.randint(0, 199)
        if random_img < 100:
            img_num = num1 * 100 + random_img
        else:
            img_num = num2 * 100 + (random_img - 100)
        img = cv2.imread(data_path + str(img_num) + ".png", 0)
        ax = plt.subplot(3, 5, 1 + i)
        ax.imshow(img, cmap="gray")
        title = str(predic[random_img]) + "(" + str(random_img) + ")"
        if label[random_img] == predic[random_img]:
            color = "b"
        else:
            color = "r"

        ax.set_title(title, fontsize=15, color=color)
        ax.set_xticks([])
        ax.set_yticks([])
    plt.show()



if __name__ == "__main__":
    num1 = 0
    num2 = 1

    training_dataset = readImg(num1, num2)
    x = trainModel(training_dataset, num1, num2)
    Y_pred, Y_true = trainModel(training_dataset, num1, num2)
    plot_result(Y_true, Y_pred, num1, num2)
```

Result:

(base) C:\Users\cghsi\Desktop\HW3>python hw1.py

Confusion matrix:

[[82 18]

 [13 87]]

Accuracy:82.435

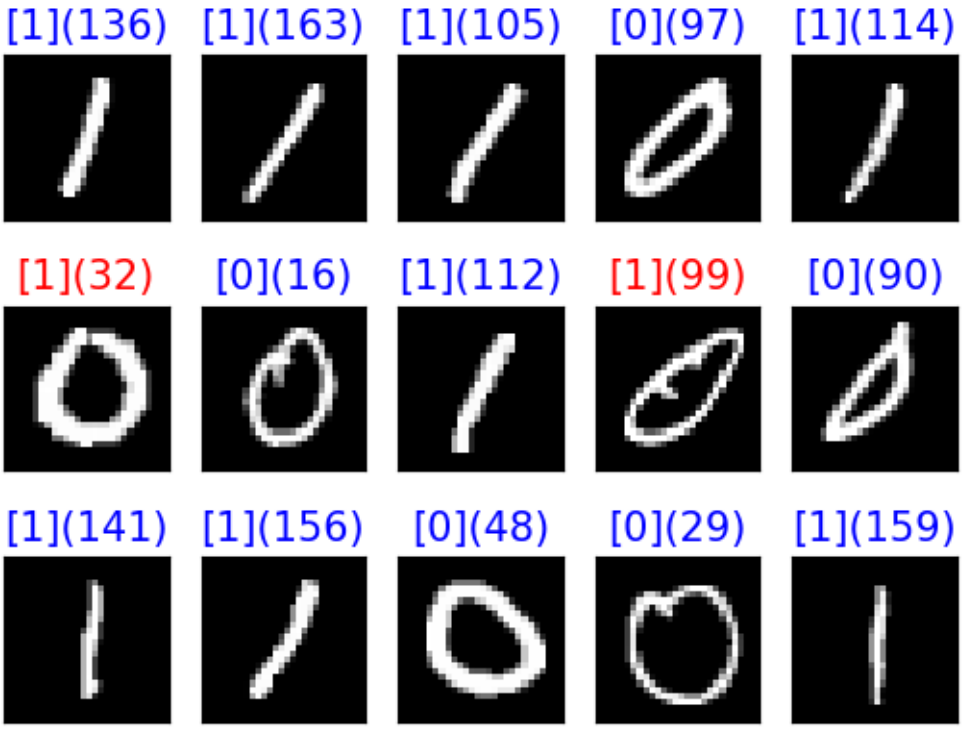

Figure 1 隨機選 15 張訓練影像顯示辨識成果，以’0’,’1’為例。